

Cell Phone Mini Challenge: Node-Link Animation Award

Animating multivariate dynamic social networks

Michael Farrugia*
University College Dublin

Aaron Quigley†
University College Dublin

ABSTRACT

This article describes the visualization tool developed for analysing a dynamic social network of phone calls, for the VAST 2008 Mini Challenge. The tool was designed to highlight temporal changes in the network, by animating different network visual representations. We also explain how animating these network representations, helped to identify key events in the mini challenge problem scenario. Finally, we make some suggestions for future research and development in the area.

Keywords: Visual analytics, Dynamic social networks, Information visualization, Graph visualization

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces (GUI); I.3.6 [Methodology and Techniques]: Interaction Techniques

1 INTRODUCTION

The motivation for developing this tool suite came when attempting to solve the VAST 2008 phone calls mini challenge [4]. The competition scenario network consisted of a set of phone calls between family members of a religious organisation living on an island. Each phone call record had the time and duration of each call and the geographic location from where the call was made. The main task was to identify key people in the organisation, and to explore how the social network structure changed over a 10 day period.

Support for visualizing network change over time is still very limited in traditional visualization or social network analysis tools. Commonly used SNA tools such as Pajek [1], have limited support for visualizing time based networks. In Pajek, the user has to manually click on a sequence of images, as in a slideshow, to animate a dynamic network. Recently, Bender-deMoll and McFarland [2], developed SoNIA, a social network image animator, in an effort to develop a framework for studying dynamic social networks using animation.

2 FEATURES

Our tool was designed to allow easy exploration and interactive animation of a dynamic network. The network can be represented at different levels of detail. At an overview level, the whole network can be visualized using a matrix representation. From this overview, interesting detailed parts of the network can be zoomed upon, and explored, using a node-link representation. Finally, the individual nodes can be studied at an instance level.

Both the coupled matrix and the node-link views can be animated to highlight the change in network structure over time. We aimed to provide the user the ability to control different animation parameters during the animation process and, to allow different aspects of the network to be highlighted during the animation. The tool was

implemented using Processing [5], which provided an ideal platform for rapid prototyping and development of visualizations and interactive features.

2.1 Matrix animation

The main purpose of the matrix representation is to highlight change in the whole network, such as an increase in activity by a group of nodes. Matrix views in current social network analysis systems, like Pajek, have restrictions on the maximum size of the network that can be displayed, in matrix mode. In our visualization we overcame this, by representing edges as small 2x2 pixel squares. This enabled us to fit the 400 node graph in a resolution of 800x800.

When animating the matrix view, the edges start appearing in the matrix, as new connections between actors are made. Each square representing an edge, can have a different colour depending on the properties of the edge. Each time the same edge occurs in the data, the colour intensity of that edge increases, while the edge colour starts fading into the background colour if it is inactive more than the decay parameter setting. Different decay functions control the way edge colours fade as they start becoming inactive. The timeline of the animation can also be dragged to a point in time, to view the activity state of the edges in the network at that time. The visualization design pattern (colour decay) helps the user develop a mental model of the temporal aspects of the network.

In the phone call network analysis, the matrix view was effective as it highlighted the introduction of very active latecomers clearly. We found that without interaction, the density of the display makes it difficult to get exact measures from the matrix view. However, with the proper highlighting of edges and some basic interaction design patterns including hover on edges, it was easy to identify where important changes in the network happened. Based on the brief experience with an animated matrix view, we believe that there is scope for further exploration of using dense pixel based matrix displays, to visualize dynamics of large graphs [3].

2.2 Node link animation

In the node-link view, the analyst can explore part of the network at any point in time. To explore the network, the user can expand, or contract a node's neighbours, by clicking on the **context menu** of each node. Node colouring was added to guide the analyst which nodes to explore further, according to the node properties. For the analysis task we used 3 node colouring options, by node (undirected) degree, node importance (based on a classification of all nodes in the network), and by node geographic location. A node can either have a fixed position, or else be allowed to move using the spring embedded layout, acting on the graph [6].

Once part of the network is extracted using the network exploration features, or the matrix zoomed into, the network can be animated. At the start of the animation all edges are removed. The edges begin appearing as they are formed as time passes. When a new edge is added between two nodes, and any of the nodes does not have a fixed position, the nodes move close together to emphasise the relationships between the nodes. This approach follows the standard force model, which emphasises co-location to emphasise association.

*e-mail: mike.farrugia@gmail.com

†e-mail: aquigley@ucd.ie

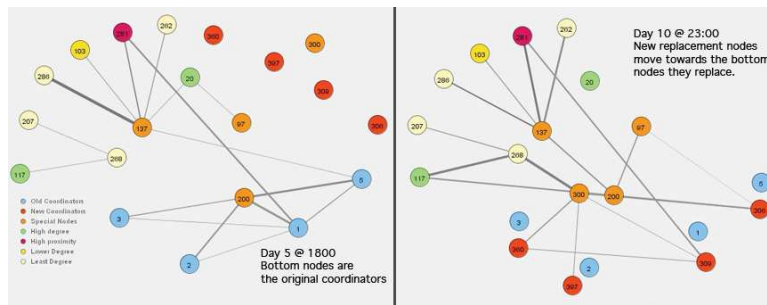


Figure 1: Frames from animation of network on days 5 and 10

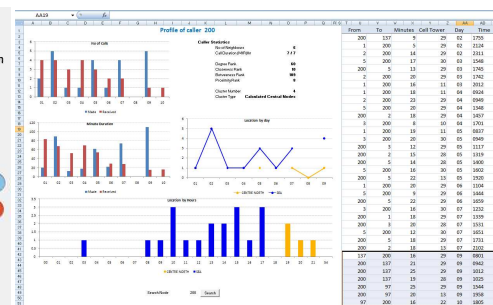


Figure 2: Detailed properties of a node in Excel

Repetitive observations of the same edge cause the edge to increase width, up to a given threshold, and then to start darkening. If a long time passes between new observations of an edge, then that edge starts fading. The time it takes for the edge to fade, can be adjusted in real time by changing the decay parameter. When the edge decays it can either decrease its prominence, or else be removed altogether, depending on the animation settings set by the analyst.

After visualising the matrix view, we got an indication that an important set of nodes started forming relationships late in the given time period. When we animated the node-link visualization, these latecomers started moving towards the nodes that were previously coordinating activity in the network. This meant that they were calling the same people that the old nodes had been calling before. With the help of this visualization, we identified the structural equivalence of the two sets of coordinating actors in the network. Figure 1 shows how the new set of nodes previously on the side in figure 1(left) moved towards the nodes they replace, at the bottom of figure 1(right).

The dataset for the mini challenge also contained the cell tower location from where each call was made. To manage this data, a geographic component was added to the visualization to explore change in each node's location over time. The geographic component uses an underlying map of hidden nodes to represent geographic areas on the map. When a call is made from a location, the node is linked to the hidden location node. The acting spring forces then draw the node close to the location. This effectively clusters the nodes based on geography. When nodes change their location, the edge with the old location is deleted, and a new one is created with the new location node. This allows the node to move from the old location, to the new location, highlighting movement of the actors in the network. From our experience, the animation speed of this visualization should be carefully controlled, to give enough time for the node to stabilise near its location. This is especially important if nodes are changing location frequently over a short span of time.

When exploring different animation parameters during analysis, we found that controlling what information is displayed, and for how long, is crucial for finding visual patterns. We provide several animation controls to allow the analyst to control the speed of the animation, and how edges fade. For instance, an edge can either fade completely after it decays, or else it can fade gradually, remaining visible but decreasing in prominence. The time it takes for an edge to fade, can also be manually adjusted in real time as this has a direct effect on how the network drawing appears. Caution must be exerted when analysing networks with fading edges, because the absence of an edge does not necessarily mean that there is no link, but that the link does not exist at a particular point in time according to the defined visual interaction decay pattern.

2.3 Interaction with Excel

Social networks don't exist in isolation, and social network information is usually supported by other non relational data. To aid the analyst in analysing a scenario, we created a dynamic report in Excel for each actor in the network. When the *view information* option is selected from a node's context menu, the profile of that node is displayed in Excel. This feature is especially useful in a multi-monitor setup, with the actor's detail displayed on one screen, and the graph in the other. All the call records made by the queried node are also displayed, in the report. We found this to be an effective way to filter through large data sets during the exploration process. Figure 2 shows the details of a node in Excel.

3 CONCLUSION AND FUTURE DEVELOPMENTS

To get a general understanding of the usefulness of the visualization system, we presented the problem scenario to some colleagues, and provided them with a summary of findings. As interest in the problem increased, each colleague asked different questions about the social interaction of actors. We used the tool to answer these adhoc questions, and were satisfied that the majority of questions were answered by using the tool. This is no substitute for formal evaluation of the tool, however we are considering the possibility of conducting user studies on different animation configurations of the visualizations.

Following our encouraging results in usefulness of the node link visualization, we plan to attempt to scale the application of this technique to larger datasets, containing more attributes. The possibility of using dense pixel displays or multi-level matrix views can also be explored further, as a way to provide a network overview with large datasets. The matrix view and the node link view can also be integrated in a single coordinated display.

There is also further scope for improving the animation features of the visualizations. Currently the user can manually control aspects of this visualization by setting parameters. There is an argument for attempting to determine these parameters automatically, to generate displays that are more likely to be of value to the user.

REFERENCES

- [1] V. Batagelj and A. Mrvar. Pajek-Program for Large Network Analysis. *Connections*, 21(2):47–57, 1998.
- [2] S. Bender-deMoll and D. McFarland. The art and science of dynamic network visualization. *Journal of Social Structure*, 7(2), 2006.
- [3] D. Keim, M. Hao, U. Dayal, and M. Hsu. Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Information Visualization*, 1(1):20–34, 2002.
- [4] V. C. C. Members. <http://www.cs.umd.edu/hcil/VASTchallenge08/index.htm>.
- [5] Processing. <http://processing.org>.
- [6] A. J. Quigley. Experience with fade for the visualization and abstraction of software views. In *IWPC '02: Proceedings of the 10th International Workshop on Program Comprehension*, page 11, Washington, DC, USA, June 2002. IEEE Computer Society.